

# Chương 1

## Ôn Tập

# **1. MỤC TIÊU**

- Ôn lại các khái niệm, các kiến thức lập trình cơ bản
- Biến toàn cục
- Biến cục bộ
- Hàm và biến toàn cục
- Tham số và hàm
- Trừu tượng hóa dữ liệu

## 2. BÀI TOÁN

- Bài toán: Viết chương trình nhập họ tên, điểm toán, điểm văn của một học sinh. Tính điểm trung bình và xuất kết quả.

### 3. BIẾN TOÀN CỤC

- **Khái niệm:** Biến toàn cục là biến được khai báo bên ngoài tất cả các hàm và được hiểu bên trong tất cả các hàm.
- Thông thường biến toàn cục được khai báo ở đầu chương trình.
- Lưu ý: Biến khai báo bên trong thân hàm main không là biến toàn cục mà là biến cục bộ của hàm main.

```
11. #include <conio.h>
12. #include <stdio.h>
13. char hoten[31];
14. int toan;
15. int van;
16. float dtb;
17. void main()
18. {
19.     printf("Nhap ho ten:");
20.     gets(hoten);
21.     printf("Nhap toan:");
22.     scanf("%d", &toan);
23.     printf("Nhap van:");
24.     scanf("%d", &van);
25.     dtb = (float) (toan + van) / 2;
26.     printf("\n Ho ten: %s",
                hoten);
27.     printf("\n Toan: %d",
                toan);
28.     printf("\n Van: %d", van);
29.     printf("\n Trung binh: %f",
                dtb);
30. }
```

## 4. BIẾN CỤC BỘ

- **Khái niệm:** Biến cục bộ là biến được khai báo và được hiểu bên trong một phạm vi nào đó của chương trình, ra khỏi phạm vi này biến không còn được biết đến nữa vì không gian bộ nhớ cấp phát cho biến được tự động thu hồi.
- Thông thường biến cục bộ được khai báo bên trong thân của một hàm hay một khối lệnh.
- Lưu ý: Một biến được khai báo bên trong thân hàm main là biến cục bộ của hàm main.

```

11. #include <conio.h>
12. #include <stdio.h>
13. void main()
14. {
15.     char hoten[31];
16.     int toan;
17.     int van;
18.     float dtb;
19.     printf("Nhap ho ten:");
20.     gets(hoten);
21.     printf("Nhap toan:");
22.     scanf("%d", &toan);
23.     printf("Nhap van:");
24.     scanf("%d", &van);
25.     dtb=(float) (toan+van) /2;
26.     printf("\nHo ten:%s",
                hoten);
27.     printf("\n Toan: %d" ,
                toan);
28.     printf("\n Van: %d" , van);
29.     printf("\n Trung binh:%f" ,
                dtb);
30. }

```

## **5. HÀM VÀ BIẾN TOÀN CỤC**

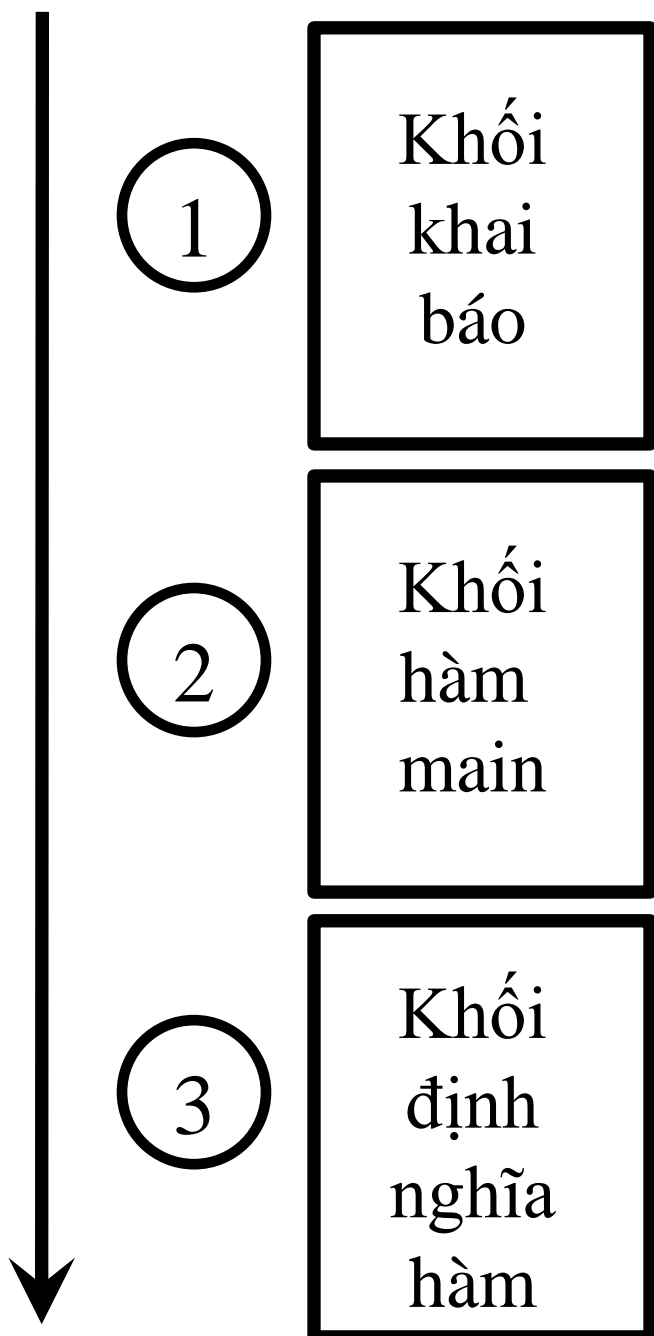
- Kiến trúc chương trình C.
- Khởi khai báo
- Khởi hàm main
- Khởi định nghĩa hàm.



## 5.1 KIẾN TRÚC MỘT CHƯƠNG TRÌNH C ĐƠN GIẢN

- Kiến trúc của một chương trình C cơ bản bao gồm 3 khối lệnh chính như sau: **khối khai báo, khối hàm main và khối định nghĩa hàm**. Ba khối lệnh này được trình bày theo thứ tự của hình vẽ bên dưới.

## 5.1 KIẾN TRÚC MỘT CHƯƠNG TRÌNH C ĐƠN GIẢN



## 5.1 KIẾN TRÚC MỘT CHƯƠNG TRÌNH C ĐƠN GIẢN

- **Khối khai báo:** chứa các khai báo hàm, khai báo biến toàn cục, khai báo sử dụng thư viện, khai báo hằng, khai báo kiểu dữ liệu...
- **Khối hàm main:** chứa duy nhất hàm main và thân hàm của nó. Trong thân hàm main chứa các lời gọi hàm cần thiết cho chương trình.
- **Khối định nghĩa hàm:** chứa các định nghĩa hàm đã được khai báo trong khối khai báo.

## 5.2 CHƯƠNG TRÌNH

- Bài toán: Viết chương trình nhập họ tên, điểm toán, điểm văn của một học sinh. Tính điểm trung bình và xuất kết quả.

## 5.2 CHƯƠNG TRÌNH

```
1. #include <conio.h>
2. #include <stdio.h>

3. char hoten[31];
4. int toan;
5. int van;
6. float dtb;

7. void Nhap();
8. void XuLy();
9. void Xuat();
```

## 5.2 CHƯƠNG TRÌNH

```
1. void main()  
2. {  
3.     Nhap();  
4.     XuLy();  
5.     Xuat();  
6. }
```

## 5.2 CHƯƠNG TRÌNH

```
11. void Xuat ()
12. {
13.     printf("Ho ten:%s", hoten);
14.     printf("Toan: %d", toan);
15.     printf("Van: %d:", van);
16.     printf("DTB: %f", dtb);
17. }
18. void XuLy()
19. {
20.     dtb=(float) (toan + van) /2;
21. }
```

## 5.2 CHƯƠNG TRÌNH

```
1. void Nhap()  
2. {  
3.     printf("Nhap ho ten:");  
4.     gets(hoten) ;  
5.     printf("Nhap toan:");  
6.     scanf ("%d", &toan) ;  
7.     printf("Nhap van:");  
8.     scanf ("%d", &van) ;  
9. }
```



## 6. THAM SỐ VÀ HÀM

- **Khái niệm: Các thông số đầu vào của một hàm được gọi là tham số của hàm.**
- Phân loại tham số: có 2 loại tham số là tham trị và tham biến.
  - + Tham trị: Không đổi.
  - + Tham biến: Thay đổi.
- Cấp phát bộ nhớ:
  - + Tham trị: Cấp phát.
  - + Tham biến: Không cấp phát bộ nhớ khi hàm được gọi thực hiện mà sử dụng bộ nhớ của đối số tương ứng.

## 6. THAM SỐ VÀ HÀM

```
1. #include <conio.h>
2. #include <stdio.h>

3. void Nhap(char [],int&,int &);
4. void XuLy(int, int, float &);
5. void Xuat(char[],int,int,float);
```

## 6. THAM SỐ VÀ HÀM

```
1. void main()  
2. {  
3.     char ht[31];  
4.     int t,v;  
5.     float tb;  
6.     Nhap(ht,t,v) ;  
7.     Xuly(t,v,tb) ;  
8.     Xuat(ht,t,v,tb) ;  
9. }
```

## 6. THAM SỐ VÀ HÀM (*tiếp*)

```
11. void Xuat(char hoten[], int  
    toan, int van, float dtb)  
12. {  
13.     printf("Ho ten: %s", hoten);  
14.     printf("Toan: %d", toan);  
15.     printf("Van: %d", van);  
16.     printf("DTB: %f", dtb);  
17. }  
  
18. void XuLy(int toan, int van,  
    float &dtb)  
19. {  
20.     dtb = (float) (toan + van) / 2;  
21. }
```

## 6. THAM SỐ VÀ HÀM (*tiếp*)

```
1. void Nhap(char hoten[30],  
           int &toan, int &van)  
2. {  
3.     printf("Nhap ho ten:");  
4.     gets(hoten) ;  
5.     printf("Nhap toan:");  
6.     scanf ("%d", &toan) ;  
7.     printf("Nhap van:");  
8.     scanf ("%d", &van) ;  
9. }
```

## 7. TRỪU TƯỢNG HÓA DỮ LIỆU

- Khái niệm: Trừu tượng hóa dữ liệu là một phương pháp tích hợp các kiểu dữ liệu đơn, các kiểu dữ liệu có sẵn nhằm mô tả, biểu diễn một khái niệm hay một đối tượng trong thế giới thực.

- Cú pháp:

```
1. struct KieuDuLieu
2. {
3.     Thành phần 1;
4.     Thành phần 2;
5.     ...
6. };
7. typedef struct KieuDuLieu
                        KIEUDULIEU;
```

## 7. TRỪU TƯỢNG HÓA DỮ LIỆU (tiếp)

```
11. #include <stdio.h>
12. #include <conio.h>
13. struct HocSinh
14. {
15.     char hoten[31];
16.     int toan;
17.     int van;
18.     float dtb;
19. };
20. typedef struct HocSinh HOCSINH;
21. void Nhap(HOCSINH &);
22. void Xuat(HOCSINH);
23. void XuLy(HOCSINH&);
```

## 7. TRỪU TƯỢNG HÓA DỮ LIỆU (tiếp)

```
1. void Nhap (HOCSINH &) ;  
2. void Xuat (HOCSINH) ;  
3. void XuLy (HOCSINH&) ;  
  
4. void main ()  
5. {  
6.     HOCSINH hs;  
7.     Nhap (hs) ;  
8.     XuLy (hs) ;  
9.     Xuat (hs) ;  
10. }
```



## 7. TRỪU TƯỢNG HÓA DỮ LIỆU (tiếp)

```
1. void Xuat (HOCSINH x)
2. {
3.     printf("Ho ten:%s",x.hoten);
4.     printf("Toan: %d",x.toan);
5.     printf("Van: %d",x.van);
6.     printf("DTB: %f",x.dtb);
7. }
8. void XuLy (HOCSINH &x)
9. {
10.    x.dtb=(float) (x.toan+x.van)/2;
11. }
```

## 7. TRỪU TƯỢNG HÓA DỮ LIỆU (tiếp)

```
1. void Nhap (HOCSINH&x)
2. {
3.     printf("Nhap ho ten:");
4.     gets(x.hoten) ;
5.     printf("Nhap toan:");
6.     scanf ("%d", &x.toan) ;
7.     printf("Nhap van:");
8.     scanf ("%d", &x.van) ;
9. }
```

## 8. ỨNG DỤNG

- **8.1 Ứng dụng 1:**  
**Viết chương trình nhập vào một phân số. Rút gọn phân số đó và xuất kết quả.**

## 8.1 ỨNG DỤNG 1

```
11. #include <stdio.h>
12. #include <conio.h>
13. struct PhanSo
14. {
15.     int tu;
16.     int mau;
17. };
18. typedef struct PhanSo PHANSO;
19. void Nhap(PHANSO &);
20. void Xuat(PHANSO);
21. void RutGon(PHANSO&);
```

## 8.1 ỨNG DỤNG 1

```
1. void Nhap (PHANSO &) ;  
2. void Xuat (PHANSO) ;  
3. void XuLy (PHANSO&) ;  
  
4. void main ()  
5. {  
6.     PHANSO ps ;  
7.     Nhap (ps) ;  
8.     RutGon (ps) ;  
9.     Xuat (ps) ;  
10. }
```

## 8.1 ỨNG DỤNG 1

```
1. void Nhap (PHANSO &x)
2. {
3.     printf("Nhap tu");
4.     scanf("%d", &x.tu);
5.     printf("Nhap mau: ");
6.     scanf("%d", &x.mau);
7. }
8. void Xuat (PHANSO x)
9. {
10.    printf("\n Tu: %d", x.tu);
11.    printf("\n Mau: %d", x.mau);
12. }
```

## 8.1 ỨNG DỤNG 1

```
1. void RutGon (PHANSO &x)
2. {
3.     int a = abs(x.tu);
4.     int b = abs(x.mau);
5.     while (a*b!=0)
6.     {
7.         if (a>b)
8.             a = a - b;
9.         else
10.            b = b - a;
11.    }
12.    x.tu = x.tu / (a+b);
13.    x.mau = x.mau / (a+b);
14. }
```

## 8.2 ỨNG DỤNG 2

- 8.2 Ứng dụng 2:  
Viết chương trình  
nhập vào tọa độ 2  
điểm trong mặt  
phẳng Oxy. Tính  
khoảng cách giữa  
chúng và xuất kết  
quả



## 8.2 ỨNG DỤNG 2

```
11. #include <stdio.h>
12. #include <conio.h>
13. struct Diem
14. {
15.     float x;
16.     float y;
17. };
18. typedef struct Diem DIEM;
19. void Nhap(DIEM &);
20. void Xuat(DIEM);
21. float KhoangCach(DIEM, DIEM);
```

## 8.2 ỨNG DỤNG 2

```
1. void Nhap (DIEM &) ;
2. void Xuat (DIEM) ;
3. float KhoangCach (DIEM, DIEM) ;

4. void main ()
5. {
6.     DIEM A, B;
7.     Nhap (A) ;
8.     Nhap (B) ;
9.     float kq=KhoangCach (A, B) ;
10.    Xuat (A) ;
11.    Xuat (B) ;
12.    printf ("\n ...: %f", kq) ;
13. }
```

## 8.2 ỨNG DỤNG 2

```
1. void Nhap (DIEM &P)
2. {
3.     printf("Nhap x: ");
4.     scanf("%f", &P.x);
5.     printf("Nhap y: ");
6.     scanf("%f", &P.y);
7. }
8. void Xuat (DIEM P)
9. {
10.    printf("\n x = %f", P.x);
11.    printf("\n y = %f", P.y);
12. }
```

## 8.2 ỨNG DỤNG 2

```
1. float KhoangCach (DIEM P, DIEM Q)
2. {
3.     |     return sqrt(
4.         |         (Q.x-P.x) * (Q.x-P.x) +
5.         |         (Q.y-P.y) * (Q.y-P.y) );
6. }
```

## 9. BÀI TẬP

- Bài 1: Viết chương trình nhập vào một phân số. Hãy cho biết phân số đó là phân số âm hay dương hay bằng không.
- Bài 2: Viết chương trình nhập tọa độ hai điểm trong không gian. Tính khoảng cách giữa chúng và xuất kết quả.

## 9. BÀI TẬP

- Bài 3: Viết chương trình nhập vào 2 phân số. Tìm phân số lớn nhất và xuất kết quả.
- Bài 4: Viết chương trình nhập vào hai phân số. Tính tổng, hiệu, tích, thương giữa chúng và xuất kết quả.

## 9. BÀI TẬP

- Bài 5: Viết chương trình nhập vào 2 số phức. Tính tổng, hiệu, tích và xuất kết quả.
- Bài 6: Viết chương trình nhập vào một ngày. Tìm ngày kế tiếp và xuất kết quả.

## 9. BÀI TẬP

- Bài 7: Viết chương trình nhập vào một ngày. Tìm ngày hôm qua và xuất kết quả.
- Bài 8: Viết chương trình nhập tọa độ 3 đỉnh A,B,C của 1 tam giác trong mặt phẳng Oxy. Tính chu vi, diện tích và tìm tọa độ trọng tâm.



## 9. BÀI TẬP

- Bài 9: Viết chương trình nhập tọa tâm và bán kính của một đường tròn. Tính diện tích và chu vi của đường tròn.

## 9. BÀI TẬP

- Yêu cầu chung:
  - + Làm tất cả các bài tập trong một workspace có tên là MSSV\_BT01.
  - + Trong workspace có 9 project tương ứng với từng bài tập.
  - + Nộp bài tập lên hệ thống Moodle theo qui định.